Package 'rLingo'

March 31, 2018

Type Package

Title R Interface to LINGO API

Version 18.0.1

Date 2018-03-02

Author Zhe Liu, Kevin Cunningham

Maintainer Zhe Liu<zliu@lindo.com>

Depends R (>= 2.14.1)

Description An R interface to LINGO API functions, which gives you the ability to access all the major features of LINGO. To download the trial version LINGO, please visit www.lindo.com/rlingo.

SystemRequirements LINGO 18.0

URL www.lindo.com/rlingo

License LGPL (>= 2.1)

Collate rLingo.R zzz.R rLingoParam.R

Repository CRAN

NeedsCompilation yes

R topics documented:

rLingo *R interface to LINGO API.*

Description

R interface to LINGO API functions. For more information, please refer to LINGO User Manual.

Details

In R interface all function names use the convention of 'r' + LINGO API function name. E.g, function rLScreateEnvLng in R corresponds to LScreateEnvLng in LINGO API.

References

LINDO SYSTEMS home page at www.lindo.com

Examples

```
#The "Simple" example, which can be found in folder "Programming Samples" of the
#installation of Lingo 15.0.

#load the package
library(rLingo)

#create Lingo enviroment object
rEnv = rLScreateEnvLng()

#open LINGO's log file
rLSopenLogFileLng(rEnv,"Simple.log")

#pass memory transfer pointers to LINGO
#define pnPointersNow
pnPointersNow = integer(1)

#@POINTER(1)
dProfit = c(100.,150.)
rLSsetDouPointerLng(rEnv, dProfit, pnPointersNow)
pnPointersNow

#@POINTER(2)
dLimit = c(100.,120.)
rLSsetDouPointerLng(rEnv, dLimit, pnPointersNow)
pnPointersNow

#@POINTER(3)
dLabor = c(1.,2.)
rLSsetDouPointerLng(rEnv, dLabor, pnPointersNow)
pnPointersNow

#@POINTER(4)
dObjective = numeric(1)
rLSsetDouPointerLng(rEnv, dObjective, pnPointersNow)
pnPointersNow
```

```
#@POINTER(5)
dStatus = numeric(1)
rLSsetDouPointerLng(rEnv, dStatus, pnPointersNow)
pnPointersNow

#@POINTER(6)
dProduce = numeric(2)
rLSsetDouPointerLng(rEnv, dProduce, pnPointersNow)
pnPointersNow

#@POINTER(7)
cComputers = "STANDARD\nTURBO\n"
rLSsetCharPointerLng(rEnv, cComputers, pnPointersNow)
pnPointersNow

#Here is the script we want LINGO to run
cScript = "SET ECHOIN 1 \n TAKE SIMPLE.LNG \n GO \n QUIT \n"

#Run the script
rLSexecuteScriptLng(rEnv, cScript)

#Close the log file
rLScloseLogFileLng(rEnv)

#check solution
dStatus
dObjective
dProduce

#delete Lingo enviroment object
rLSdeleteEnvLng(rEnv)
```

`rLSclearPointersLng` *Clears out the list of @POINTER() pointers to user memory transfer areas established through calls to* `rLSsetDouPointerLng`, `rLSsetIntPointerLng`, *or* `rLSsetCharPointerLng`.

Description

R interface function for LINGO API function `LSclearPointersLng`. For more information, please refer to LINGO User Manual.

Usage

```
rLSclearPointersLng(env)
```

Arguments

`env`	A LINGO environment object.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLScloseLogFileLng` *Closes LINGO's log file that was opened previously by a call to* `rLSopenLogFileLng`.

Description

R interface function for LINGO API function `LScloseLogFileLng`. For more information, please refer to LINGO User Manual.

Usage

```
rLScloseLogFileLng(env)
```

Arguments

`env`	A LINGO environment object.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLScreateEnvLicenseLng` *Create a LINGO environment object by specifying a license key.*

Description

R interface function for LINGO API function `LScreateEnvLicenseLng`. For more information, please refer to LINGO User Manual.

Usage

```
rLScreateEnvLicenseLng(pcLicenseKey)
```

Arguments

`pcLicenseKey`	A text string containing a LINGO license key.

Value

If successful, `rLScreateEnvLicenseLng` returns a LINGO environment object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLScreateEnvLng rLSdeleteEnvLng

---

rLScreateEnvLng *Create a LINGO environment object.*

Description

R interface function for LINGO API function LScreateEnvLng. For more information, please refer to LINGO User Manual.

Usage

```
rLScreateEnvLng()
```

Value

If successful, rLScreateEnvLng returns a LINGO environment object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSdeleteEnvLng rLScreateEnvLicenseLng

---

rLSdeleteEnvLng *Deletes a LINGO environment object previously created through a call to* rLScreateEnvLng.

Description

R interface function for LINGO API function LSdeleteEnvLng. For more information, please refer to LINGO User Manual.

Usage

```
rLSdeleteEnvLng(env)
```

Arguments

env	A LINGO environment object.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSexecuteScriptLng` *Main workhorse of the LINGO DLL that processes LINGO command scripts.*

Description

R interface function for LINGO API function `LSexecuteScriptLng`. For more information, please refer to LINGO User Manual.

Usage

```
rLSexecuteScriptLng(env,pcScript)
```

Arguments

env	A LINGO environment object.
pcScript	A character string containing a LINGO command script.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSopenLogFileLng` *Creates a file for LINGO to write a log to while processing your script.*

Description

R interface function for LINGO API function `LSopenLogFileLng`. For more information, please refer to LINGO User Manual.When using this function, make sure Lingo have write permission to the folder where the Lingo log file will be sent.

Usage

```
rLSopenLogFileLng(env,pcLogFile)
```

Arguments

env	A LINGO environment object.
pcLogFile	A string containing the pathname for the log file.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLScloseLogFileLng

---

rLSparam *LINGO API Parameters.*

Description

For more information, please refer to LINGO User Manual.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsetCharPointerLng *Passes a list of character type memory pointers to LINGO.*

Description

R interface function for LINGO API function LSsetCharPointerLng. For more information, please refer to LINGO User Manual.

Usage

```
rLSsetCharPointerLng(env,pacPointer,pnPointersNow)
```

Arguments

env	A LINGO environment object.
pacPointer	A string array whose memory location will be used by an instance of @POINTER().
pnPointersNow	A one dimensional integer array in which LINGO returns the current number of entries in the @POINTER() pointer list.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSclearPointersLng

---

rLSsetDouPointerLng *Passes a list of double type memory pointers to LINGO.*

Description

R interface function for LINGO API function LSsetDouPointerLng. For more information, please refer to LINGO User Manual.

Usage

```
rLSsetDouPointerLng(env,padPointer,pnPointersNow)
```

Arguments

env	A LINGO environment object.
padPointer	A double type array whose memory location will be used by an instance of @POINTER().
pnPointersNow	A one dimensional integer array in which LINGO returns the current number of entries in the @POINTER() pointer list.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSclearPointersLng

rLSsetIntPointerLng *Passes a list of integer type memory pointers to LINGO.*

Description

R interface function for LINGO API function LSsetIntPointerLng. For more information, please refer to LINGO User Manual.

Usage

```
rLSsetIntPointerLng(env,panPointer,pnPointersNow)
```

Arguments

env	A LINGO environment object.
panPointer	A integer type array whose memory location will be used by an instance of @POINTER().
pnPointersNow	A one dimensional integer array in which LINGO returns the current number of entries in the @POINTER() pointer list.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSclearPointersLng

Index